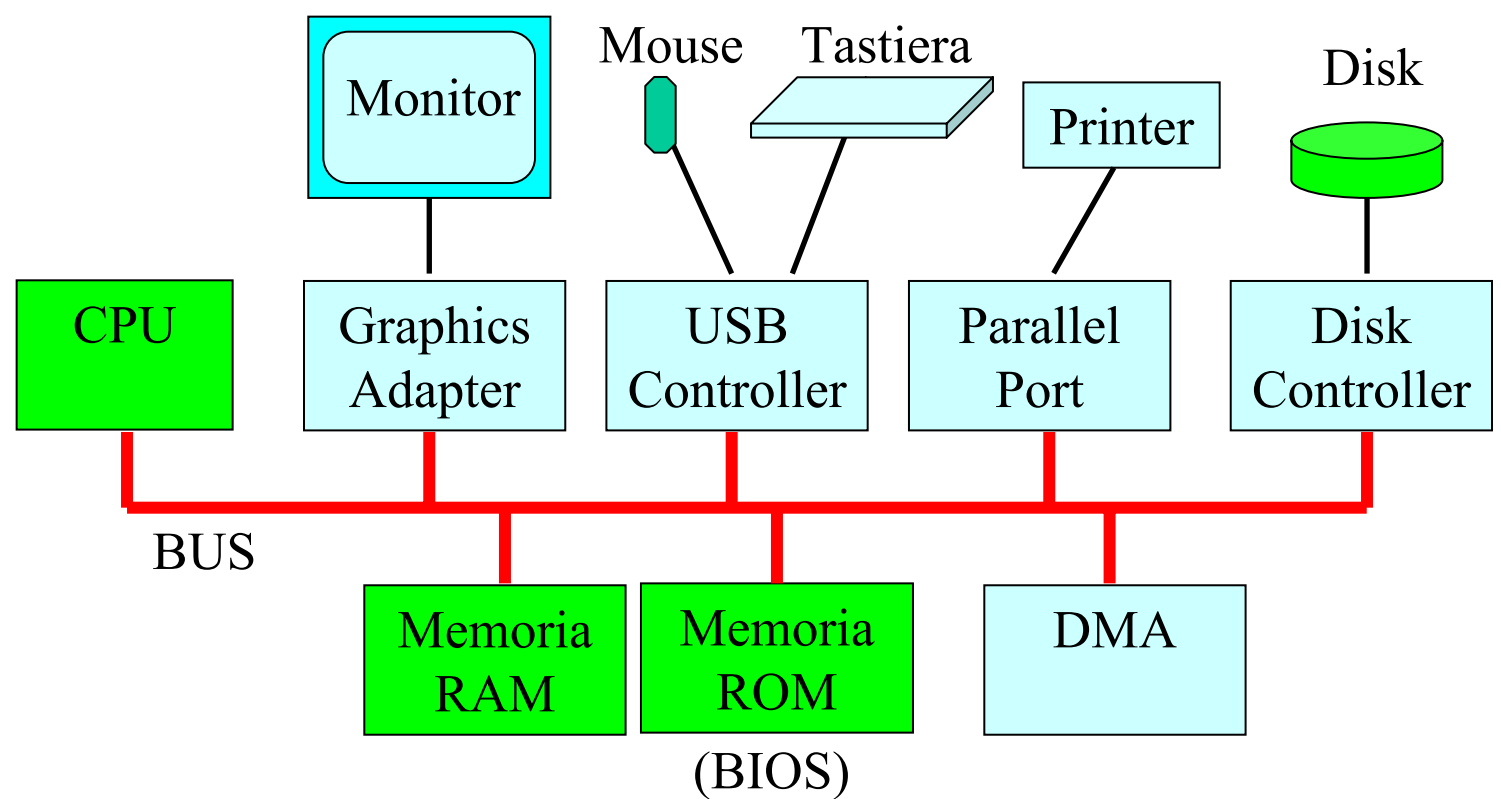


Hardware di un Computer



CPU esegue istruzioni, effettua calcoli, controlla input/output, coordina spostam. dati.

BUS trasferisce i dati tra la memoria e gli altri dispositivi.

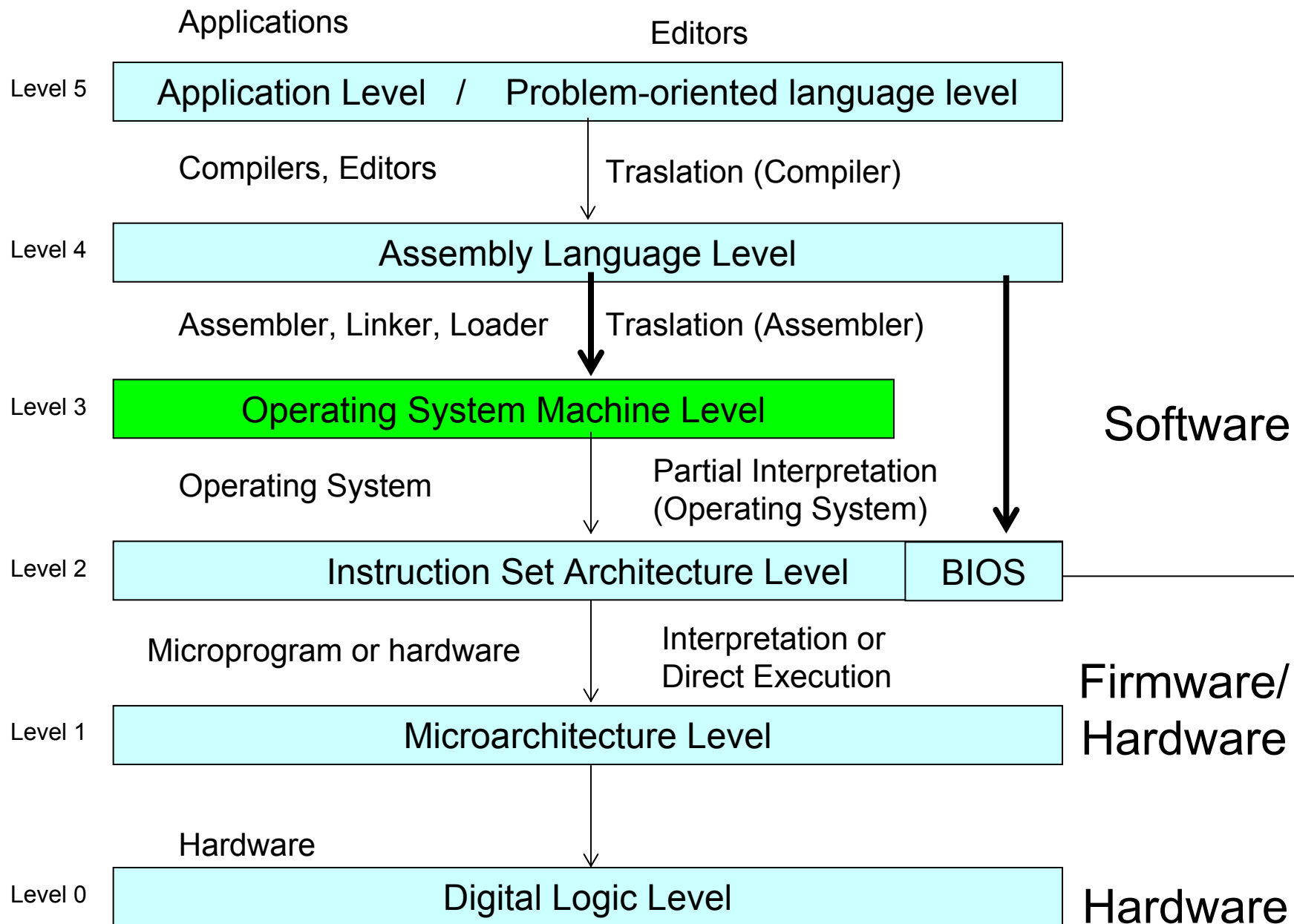
Memoria RAM (volatile) mantiene programmi (seq. di istruzioni) quando devono essere eseguiti, e i dati che i programmi usano, ma solo finché il computer è acceso.

Memoria ROM (permanente) contiene il BIOS, le istruzioni per fornire i servizi base, usate dalla CPU anche nel momento dell'accensione del computer (il BOOT).

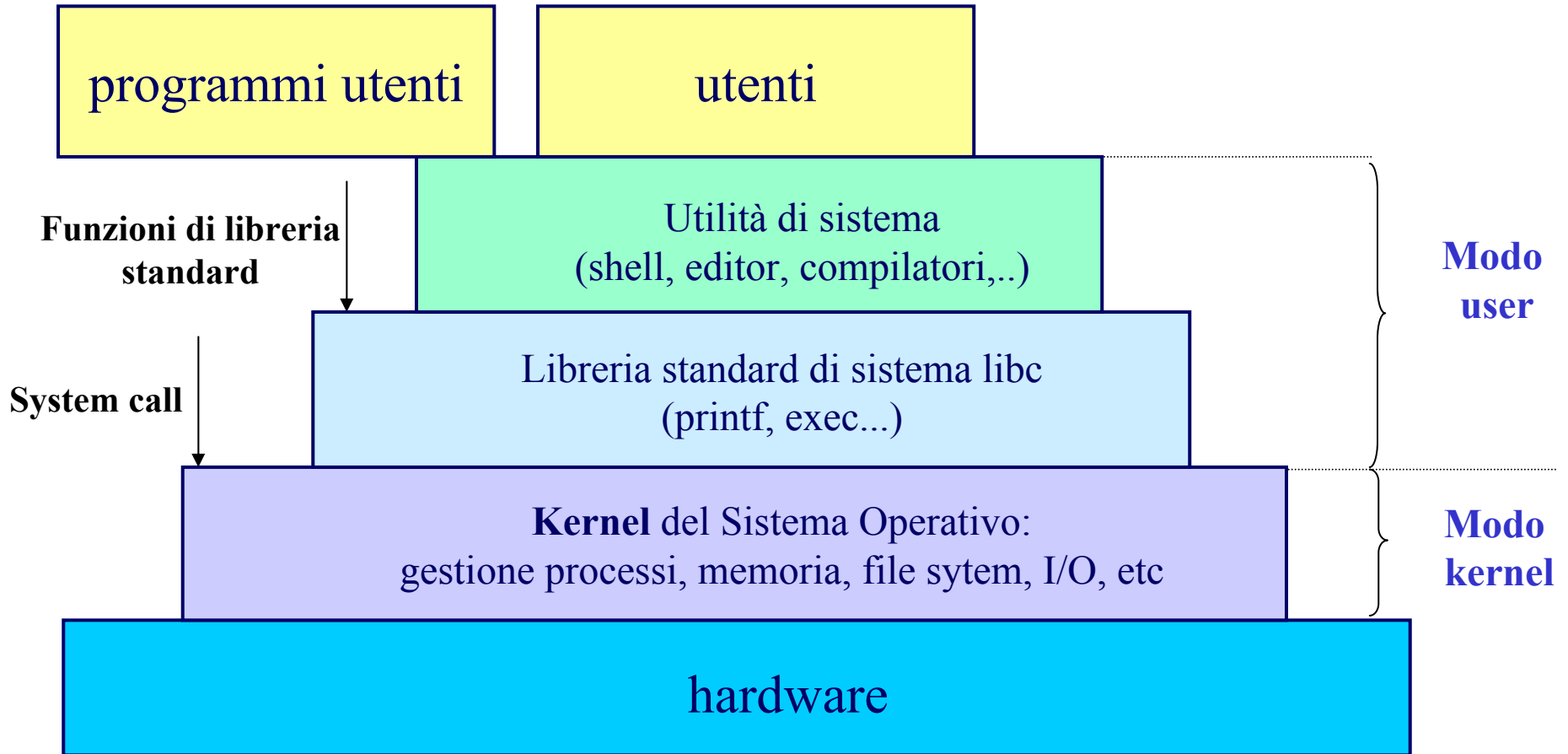
Dischi mantengono enormi quantità di programmi e dati, anche dopo lo spegnimento del computer.

Sono inoltre presenti alcune periferiche di **I/O** (Input/Output).

Collocazione architetturale del Sistema Operativo



Interazioni con il Sistema Operativo



Il sistema operativo mette a disposizione diversi servizi:

Gestione della Protezione:

- protezione della CPU - sfrutta la capacità della CPU di operare in due diverse modalità:
 - **modalità kernel**, che permette di utilizzare tutte le istruzioni ISA, tutti gli indirizzi di memoria, tutti i registri, tutte le periferiche.
 - **modalità utente**, che limita le istruzioni eseguibili, l'accesso alla memoria ai registri e alle periferiche.
- protezione della Memoria.

Gestione della CPU:

- scheduling (quale processo deve usare la CPU e per quanto?)

Gestione della Memoria:

- Indirizzamento, Segmentazione, Protezione, Memoria Virtuale, Paging.

Gestione dell' Input/Output:

- Periferiche I/O (tastiera, video, rete, dischi), Storage (organizzazione del filesystem).

Chiamate di Sistema (System Calls), Librerie di sistema, programmi di utilità:

- Servizi messi a disposizione in diverse modalità. Le system call sono disponibili mediante chiamate ad interrupt effettuabili in assembly.

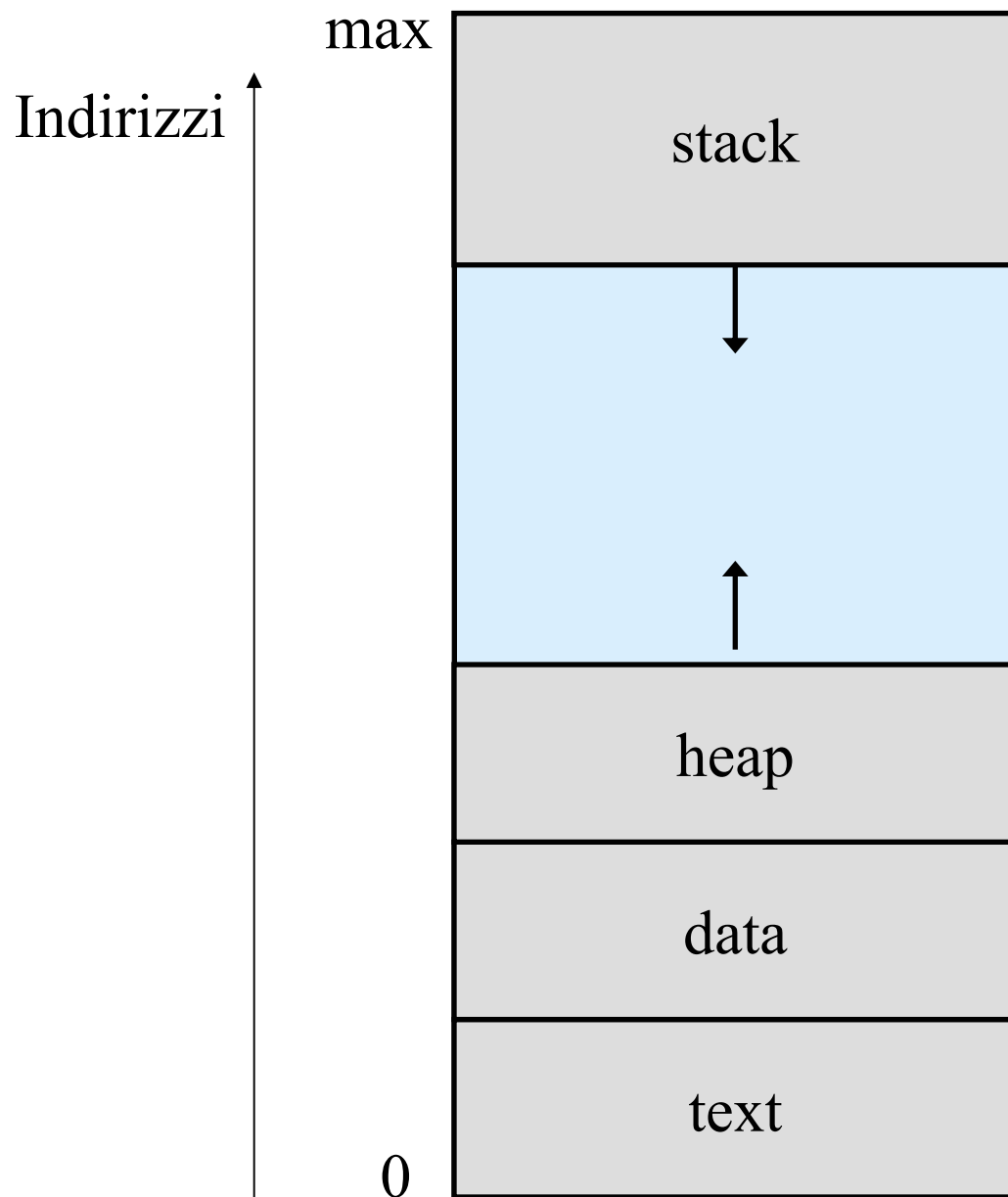
Processi

Un processo è l'istanziamento in memoria di un programma eseguibile.

Quando un programma viene lanciato, il **loader** (che fa parte del sistema operativo) :

- crea i segmenti di memoria per il processo costruendo la tabella con le informazioni sui diversi segmenti,
- carica in memoria il codice eseguibile (sezione text) del programma,
- carica in memoria la sezione Data,
- crea lo stack utente per l'esecuzione delle funzioni del programma,
- crea lo stack di sistema per l'esecuzione delle system calls che verranno invocate dal programma utente,
- copia eventuali argomenti passati a riga di comando mettendoli a disposizione del main
- infine ordina alla CPU di eseguire la prima istruzione eseguibile corrispondente al main del programma.

Organizzazione Memoria usata dai Processi in modalità utente



Il sistema operativo dipende strettamente dall'hardware su cui opera e dal livello ISA (Instruction Level Architecture) che questo mette a disposizione.

I processori moderni hanno caratteristiche comuni:

Modalità di protezione (modo kernel, modo utente)

ALU, unità aritmetico-logica.

Registri general purpose (di uso generale)

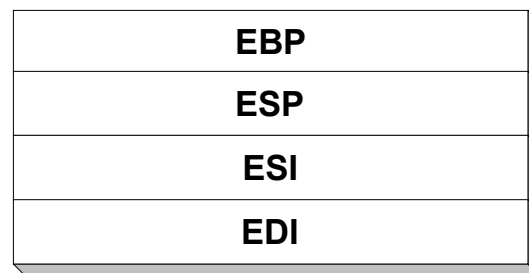
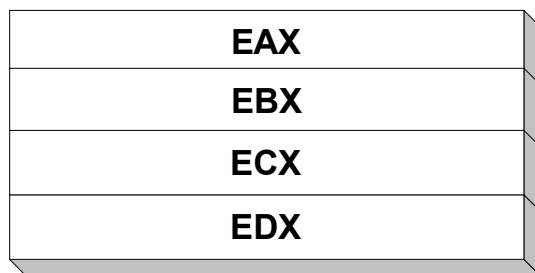
Registri Specializzati:

- Instruction Pointer (IP) serve a formare l'indirizzo in memoria (detto Program Counter (PC)) della prossima istruzione da eseguire.
- Program Status (PS) detto anche Registro di Stato (Status Register).
- Stack Segment Register (SS). Punta all'inizio dello stack.
- Code Segment Register (CS). Punta all'inizio della sezione text, il codice eseguibile.

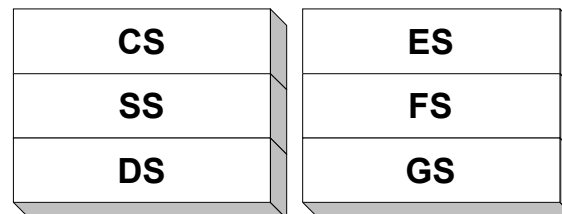
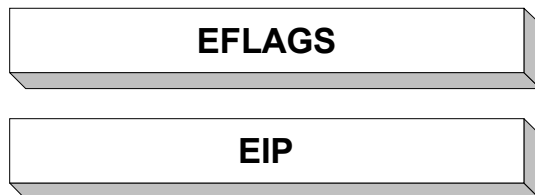
Registri disponibili ai Programmi nei processori di tipo IA-32 (386, Pentium, ..)

- Registers are high speed memory inside the CPU
 - Eight 32-bit general-purpose registers
 - Six 16-bit segment registers
 - Processor Status Flags (EFLAGS) and Instruction Pointer (EIP)

32-bit General-Purpose Registers

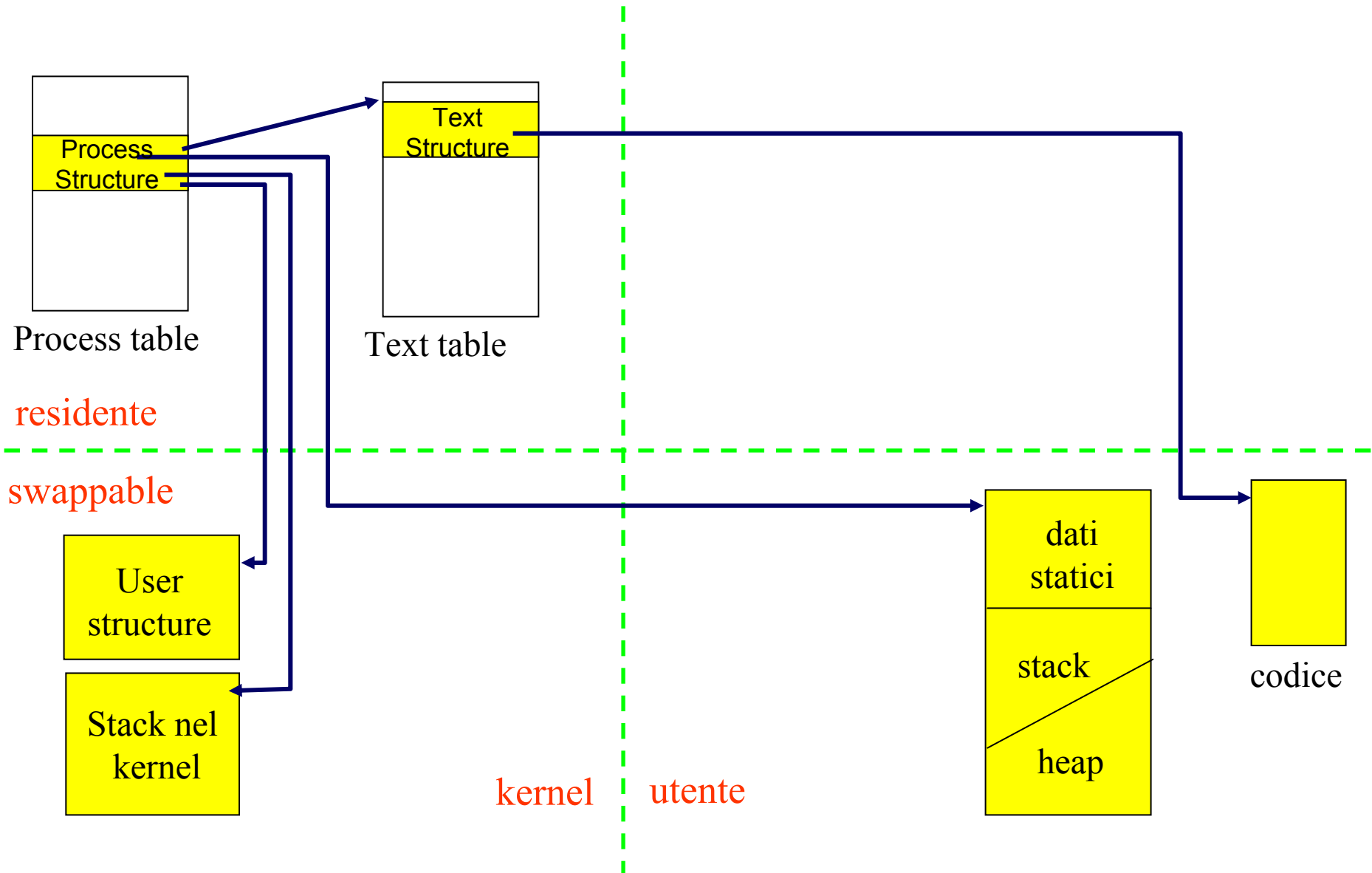


16-bit Segment Registers



- Inoltre, **Registro di Stato (Status Register)**, non disponibile ai programmi, definisce modalità di funzionamento, utente o kernel

Immagine in Memoria dei Processi (utente & kernel)



- Notare la presenza di uno stack per le funzioni utente nello spazio utente e di uno stack per l'esecuzione delle system calls in spazio kernel.